

What is claimed is:

1. A computer system for supporting communication between a plurality of users and at least one application server comprising:

an application program for receiving a client request from a client program executing on a computer associated with at least one of the users;

a client interface program for communicating messages between said client program and said application service program;

a service manager bean coupled to said application service program for creating and returning to said client program a handle to a functional bean appropriate to the client request;

a data store interface for coupling said application service program to a data storage system; and

memory coupled to said application service program, said memory for queuing customer requests and to service the queued customer requests in accordance with the code contained in the functional bean and for interfacing with the data storage system via the data store interface.

2. The computer system of claim 1 wherein the functional bean is accessible by a program running on the client via an EJBObject.

3. The computer system of claim 1 wherein the functional bean is a modified entity bean.

4. The computer system of claim 1 wherein the functional bean is configured to provide transactional persistence to a client transaction.

5. The computer system of claim 1 wherein the client is web-based.

6. The computer system of claim 1 wherein the client is an application, an applet, a servlet or a JSP that can communicate with an EJBObject's remote interface using RMI over TCP/IP or IOP.

1 7. A computer system comprising a plurality of sets of functional beans each set
2 comprising at least one functional bean assigned to perform a particular business method, the
3 computer system comprising:

4 a microprocessor;

5 a memory device coupled to the microprocessor;

6 a service manager program coupled to the memory device and configured to receive a
7 number of requests from at least one of a plurality of types of transactions from a plurality of
8 clients;

9 at least one of a plurality of resources;

10 a load-sharing program coupled to the service manager program and configured to
11 create instances of functional beans based a criterion;

12 the service manager program configured to obtain a handle to an instance of a
13 functional bean based on a type of transaction requested by a client; and

14 the service manager program configured to return the handle to the client, wherein the
15 client is configured to use the handle to interact with the functional bean to execute a
16 business method.

1 8. The computer system of claim 7 wherein the criterion is based on the number
2 of resources.

1 9. The computer system of claim 7 wherein the criterion is based on the number
2 of requests from the plurality of clients.

1 10. Computer-processor-executable software code stored in a computer-readable
2 memory, said code comprising:

3 instructions to direct a computer processor to receive a first request from a client for a
4 handle to a particular type of functional bean, said functional bean, which comprises code to
5 execute a particular business function;

6 instructions to direct the computer processor to create an instance of a functional bean
7 of the particular type requested;

8 instructions to direct the computer processor to obtain a handle to said instance of

9 said functional bean; and
10 instructions to direct the computer processor to transmit to the client said handle to
11 said instance of said functional bean,
12 wherein the computer processor, responsive to a second request from the client
13 enables the client to execute code comprised in the functional bean to accomplish the
14 particular business function.

1 11. The code of claim 10, further comprising instructions to receive the first
2 request and the second request from the client via a computer network.

1 12. The code of claim 10, further comprising instructions to create a number of
2 instances of functional beans of the particular type requested, said number being dependent
3 on availability of resources.

1 13. The code of claim 10, further comprising instructions that allow a functional
2 bean to instantiate a second functional bean of a second type in order to execute the business
3 logic contained in the second functional bean instance.

1 14. The code of claim 10, further comprising:
2 instructions that allow a client to create a session with an instance of a session
3 Enterprise JavaBean; and
4 instructions that allow the session Enterprise JavaBean to provide access to invoke
5 the business methods contained in the functional bean.

1 15. The code of claim 10, further comprising:
2 instructions to instantiate an entity Enterprise JavaBean, said entity Enterprise
3 JavaBean containing logic that maps a particular entity, and methods to perform actions on
4 the particular entity; and
5 instructions to invoke the methods contained in the entity Enterprise JavaBean from
6 the business methods contained in the functional bean.

1 16. The code of claim 10, wherein the instructions to create an instance of a
2 functional bean of the particular type requested further comprise:
3 instructions to verify if required system resources are available.

1 17. The code of claim 16 further comprising:
2 if required system resources are not available, instructions to direct the computer processor to
3 an existing instance of a functional bean of the particular type requested.

1 18. A method of using an object-oriented middleware component to create a
2 functional bean, the method comprising the steps of:
3 deriving a class with no data elements from the object-oriented middleware
4 component; and
5 adding a set of computer-executable instructions to the derived class,
6 wherein said set of computer-executable instructions is configured to provide transactional
7 access to a pool of scarce system resources allowing client requests to be queued on EJB
8 instances taken from the pool.

1 19. The method of claim 18, wherein the middleware component is an entity
2 Enterprise JavaBean.